

# Réalité augmentée

QRCode

Unity/Vuforia

**Christophe Vestri**

Le mardi 22 janvier 2019

# Plan du cours

- 8 janvier : Réalité augmentée intro, Unity/Vuforia et projet
  - 17 janvier: Construction RA + Notions 2D + exercices JS
  - 22 janvier: QRCode, Unity/vuforia et projet
  - 29 janvier: Vision par ordinateur et RA (openCV – exo C++) et projet
  - 5 février : Résumé et présentation des Projets
- 
- **Suite: Cours Cartographie/JS/AR/VR**

# **QR code**

**Christophe Vestri**

# Plan Cours 3

- QRCode
  - Introduction
  - Comment les lire
  - Création/tests
- Projets et Unity cont.

# QR code?

- **"QR"** signifie **Quick Response**
- Nécessite 1 lecteur de QR code = 1 application, 1 programme
- Il est né au Japon,
- Qui l'a inventé ?
  - [DENSO Wave](#)
  - En 1994,
  - publication en licence libre en 1999
- Il l'ont inventé pour faire quoi ?
  - Suivre les pièces automobiles dans les usines Toyota

# QR Code, un code

## Pourquoi un code ?

Pour lire rapidement et automatiquement (sans saisie) des données complexes

- > gagner du **temps**
- > **réduire les erreurs** de saisie



Pas de code  
Saisie manuelle



Codes barres  
Capture laser

pour professionnels

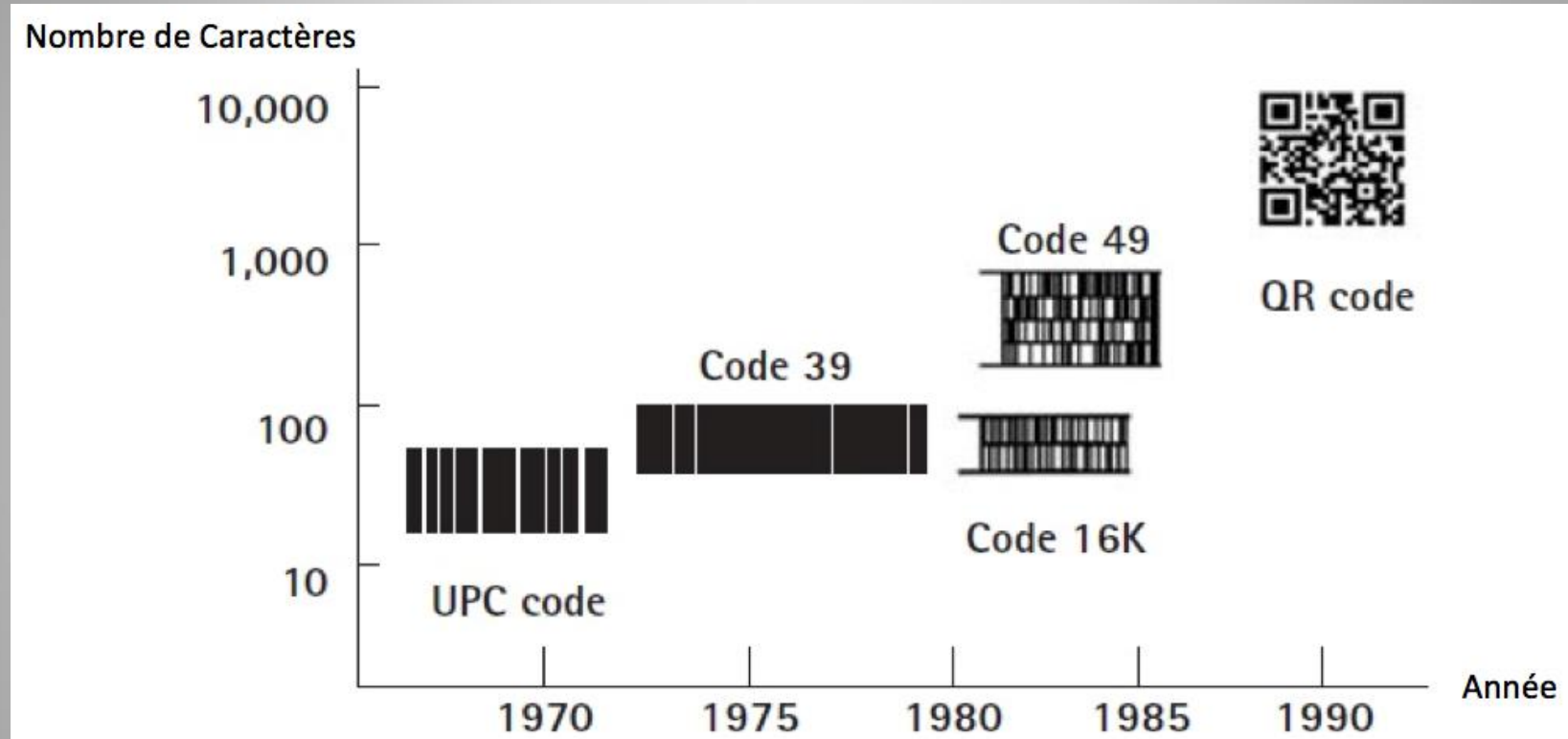


Codes 2D (dont QR code)  
Capture avec un téléphone

Pour M. Toutlemonde



# Du 1D au 2D



# Comparaison 1D – 2D

	<b>CODE BARRE</b>	<b>CODE QR</b>
ÉLÉMENTS CONSTITUTIFS	Barres	Carré blanc ou noir = <b>Module</b>
INFORMATION COMMUNIQUÉE PAR	Largeur des Barres <u>ET</u> des Espaces	Arrangement Spatial de ses Modules
DIMENSION(S)	<b>1</b> (horizontal – Lignes Uniquement)	<b>2</b> (horizontal <u>ET</u> vertical – Lignes et Colonnes)

**Modules Foncé ou Clair = Bit** ➔ Représente "0" OU "1"  
(cf. Codage)

= Instance Spécifique du Code ➔ Lisible par les machines  
MAIS aucun sens pour l'oeil nu...



# Reconnaitre les codes 2D



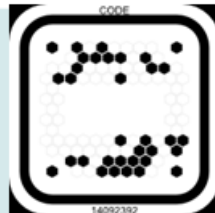
3 carrés aux coins  
**QR code**



2 barres de côté  
**Datamatrix**  
ou **Flashcode**



1 carré central  
**Aztec**



Alvéoles  
**Beetag**



Lignes latérales  
**PDF 417**



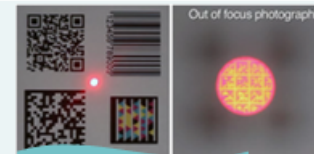
Rond central  
**maxi code**



triangles  
**Microsoft Tag**



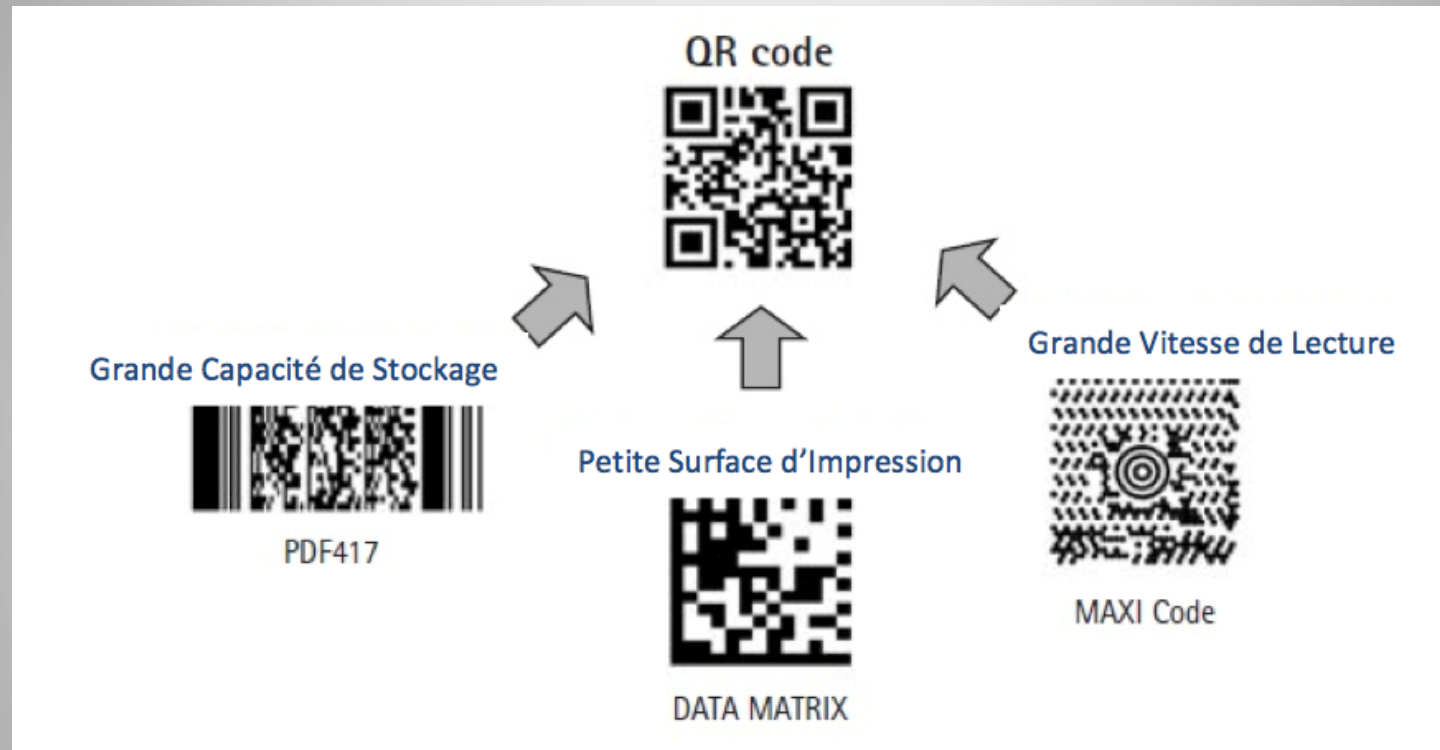
Tout rond  
**Shotcode**



nouveau

Rond, lumineux  
**Bokode**

# QR code



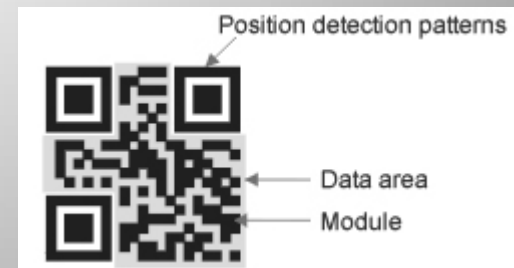
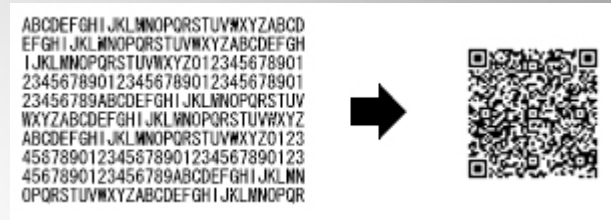
MAIS code QR = leader car regroupe points forts des autres

**Breveté** initialement **MAIS** actuellement dans le **domaine public**

# Avantages

Avantages / code barre :

- Grande capacité
- Taille d'impression réduite
  - Distance/10
- Résistant à l'usure
  - Accepte 30% de code manquant
- Lisible dans tous les sens



# QR Code pour Coder

- Dans un QR code on peut coder :
  - 1 texte alpha numérique (un message, un numéro de série, une date de création, une DLC...)



# QR Code pour Coder

- Dans un QR code on peut coder :
  - 1 URL = 1 adresse web (site web, page web, vidéo, musique...)



# QR Code pour Coder

- Dans un QR code on peut coder :
  - 1 contact complet (nom, téléphone, adresse...)



# QR Code pour Coder

- Dans un QR code on peut coder :
  - 1 action spécifique (envoyer sms/email, télécharger une app, afficher un plan, obtenir un code wifi)



# Applications

## Packaging

## Publicité





# Applications

## Publicité



# Applications

Jeux





# Applications



Carte visite

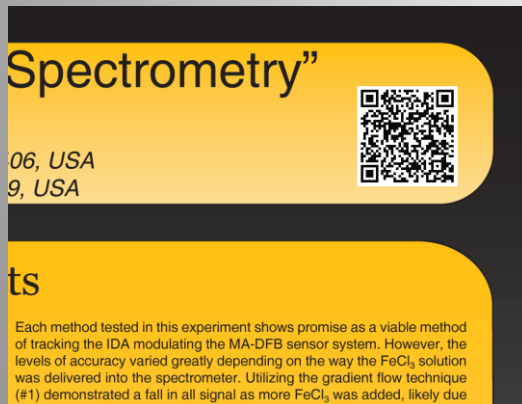
CV



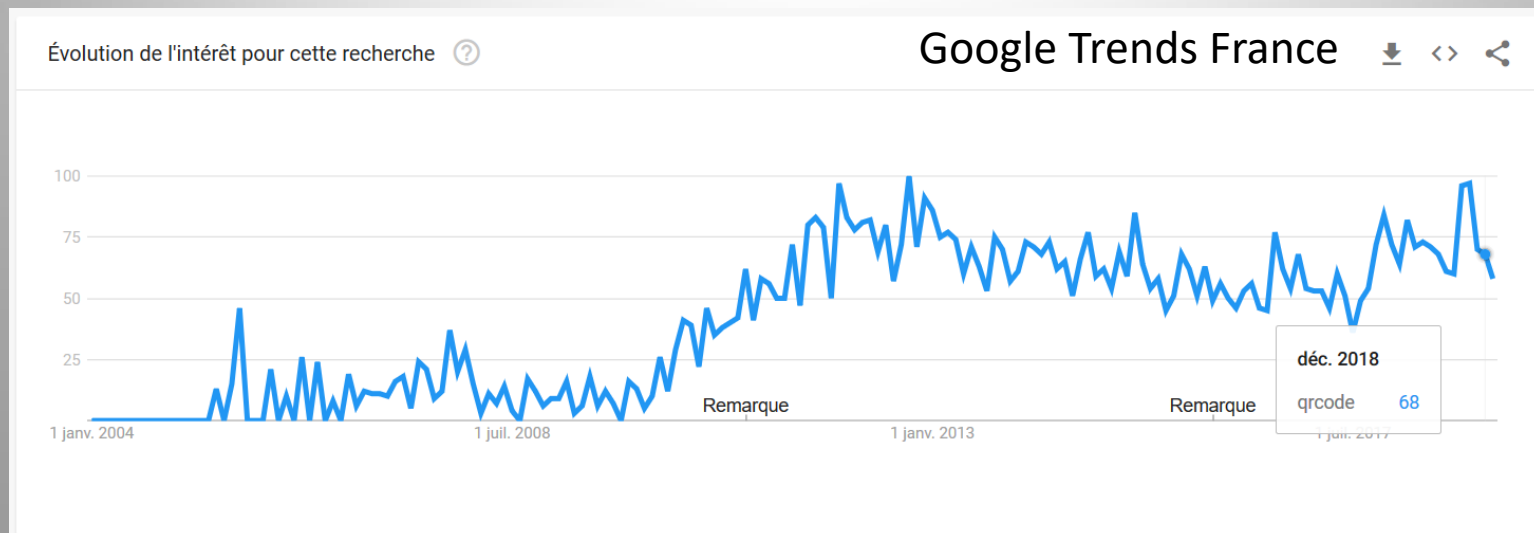
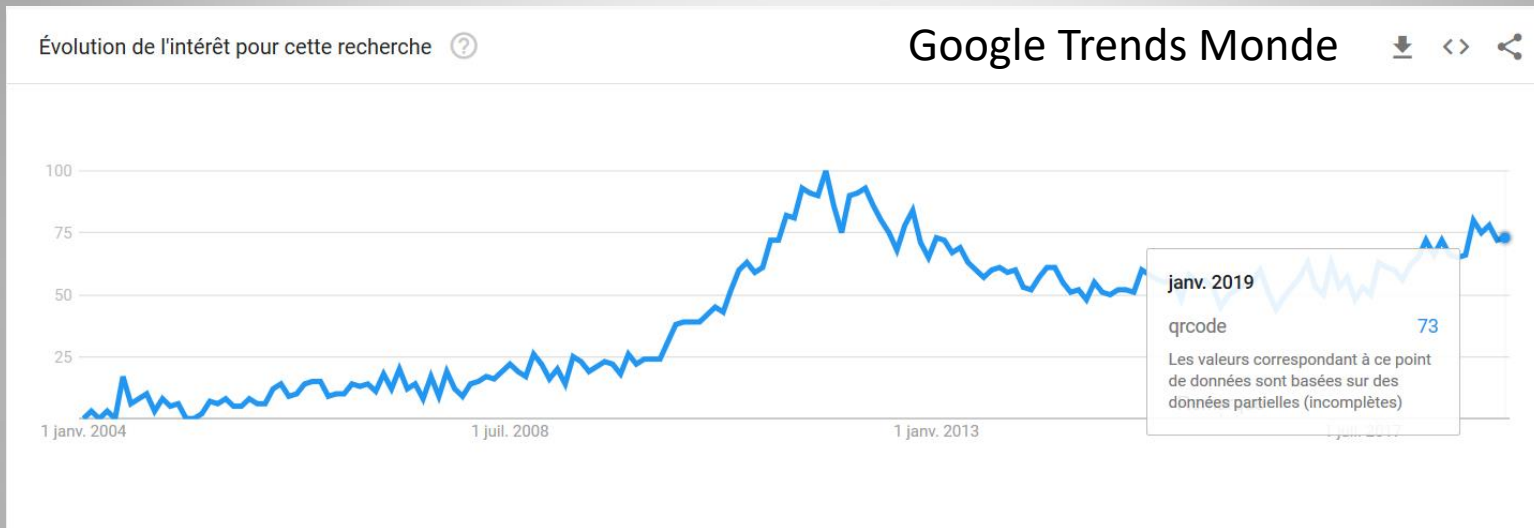
Coupons

Livres

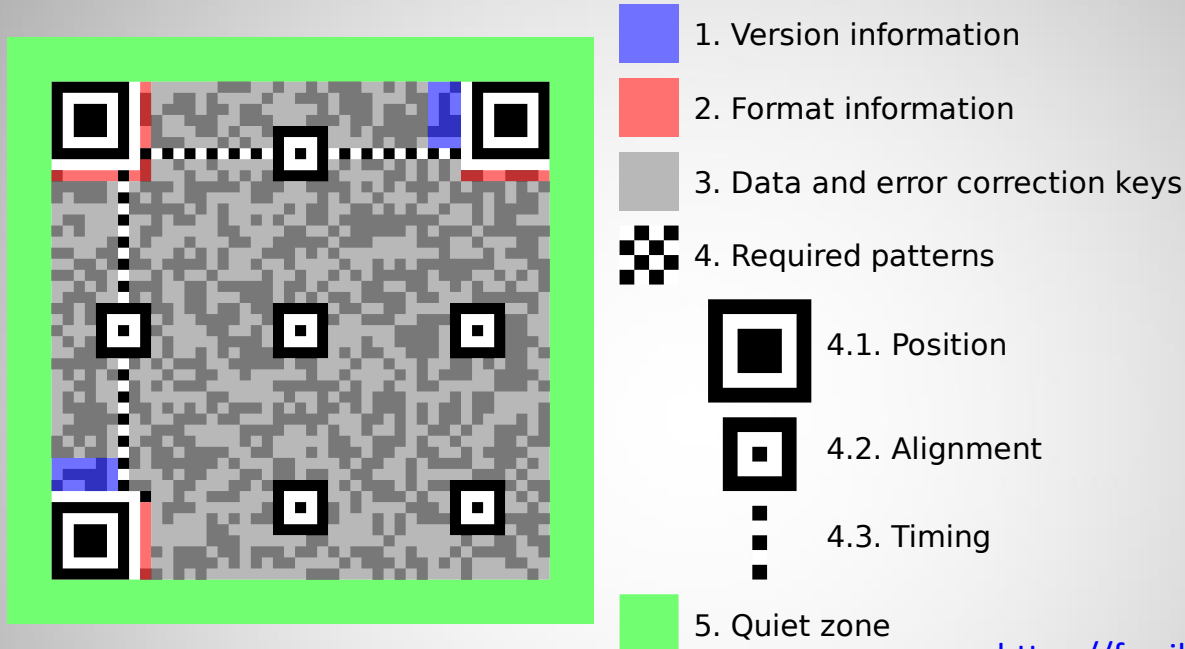
Poster



# QR Code, un code



# QR Code format



[https://fr.wikipedia.org/wiki/Code\\_QR](https://fr.wikipedia.org/wiki/Code_QR)

## Capacité de stockage

Caractères numériques : max 7 089

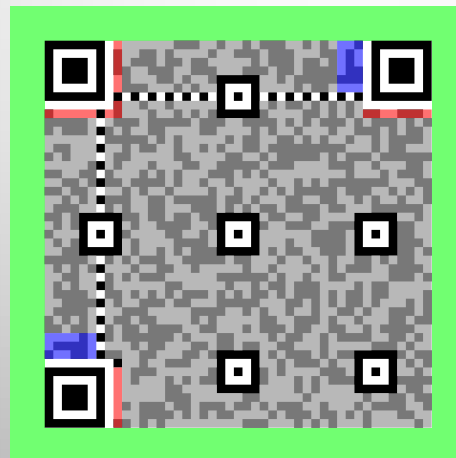
Caractères alphanumériques : max 4296

Binares (8-bits) : max 2953 octets

Kanji/Kana : max 1817 caractères

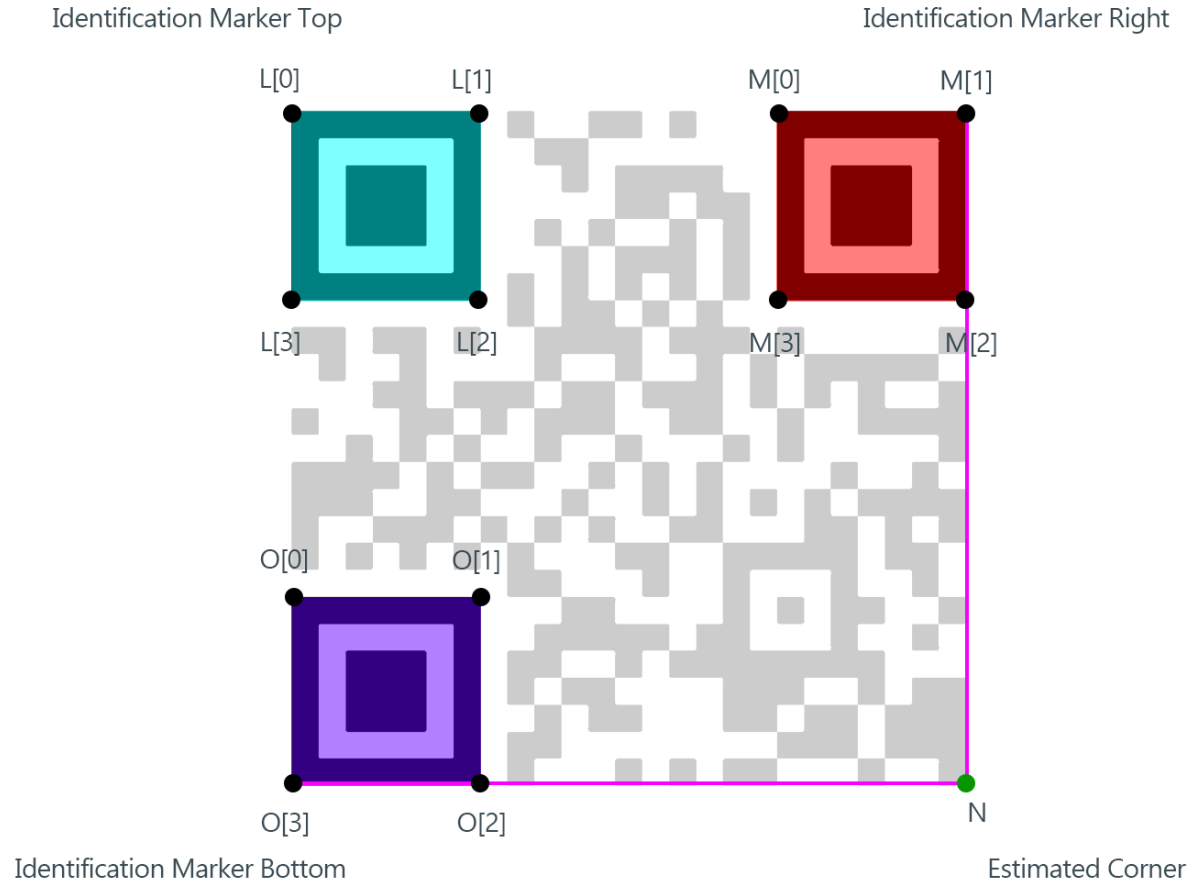
# Détection du QR Code

- Etapes:
  1. Détecter les marqueurs
  2. Corriger la perspective
  3. Lire le format et informations
  4. Décoder les données



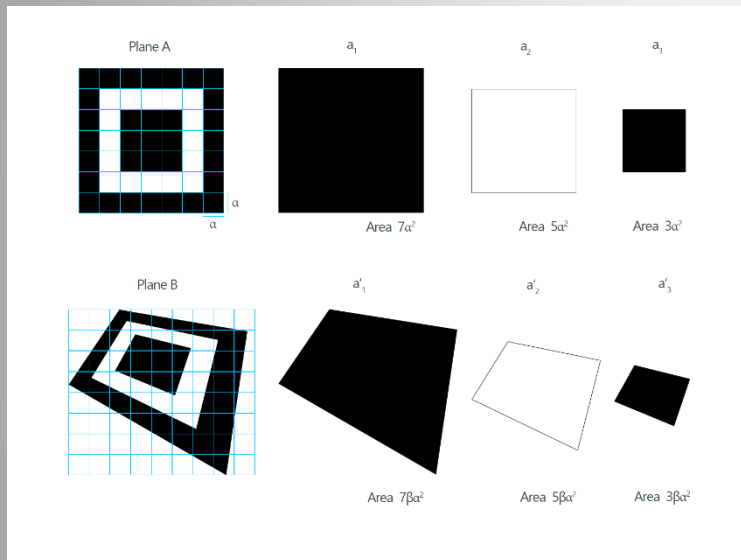
- 1. Version information
- 2. Format information
- 3. Data and error correction keys
- 4. Required patterns
  - 4.1. Position
  - 4.2. Alignment
  - 4.3. Timing
- 5. Quiet zone

# Détecter les marqueurs

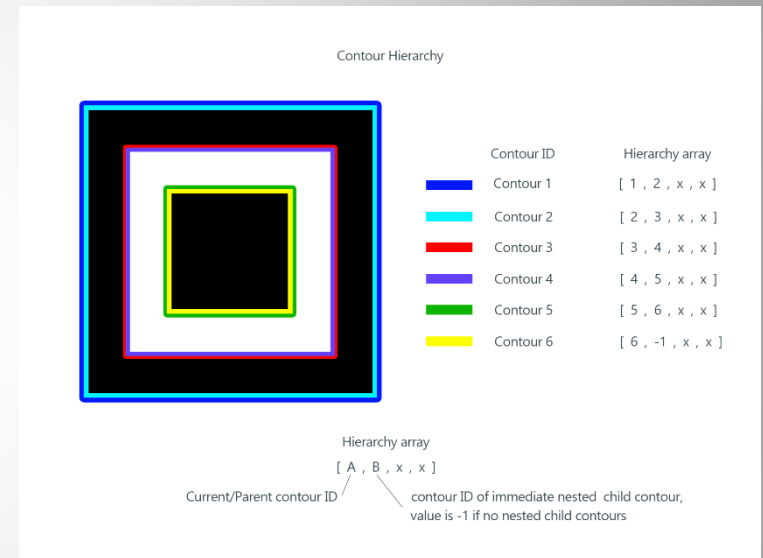


# Détecter les marqueurs

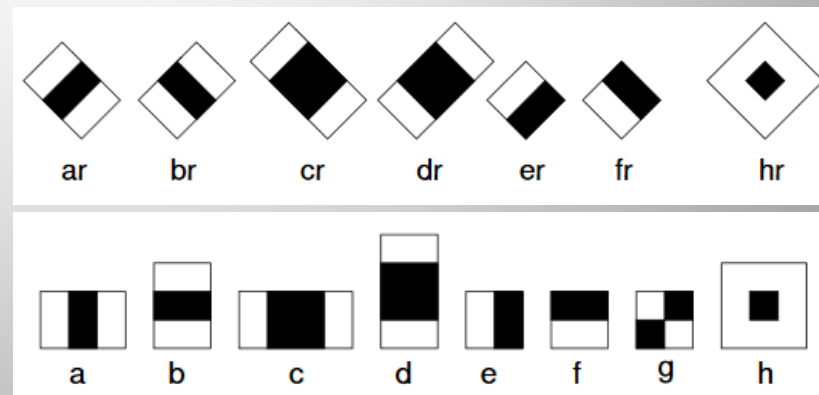
Utiliser les proportions



Utiliser les contours



Haar wavelet





# Détecter les marqueurs



Image



Binarization



Détection contour



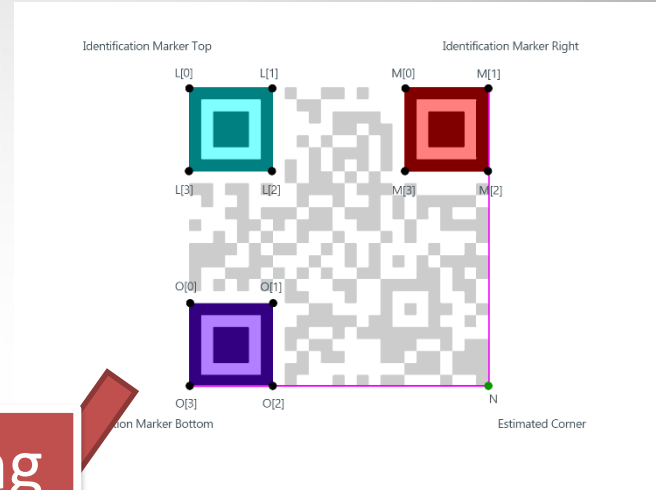
Détection marqueurs

# Corriger la perspective



Marqueurs

Warping

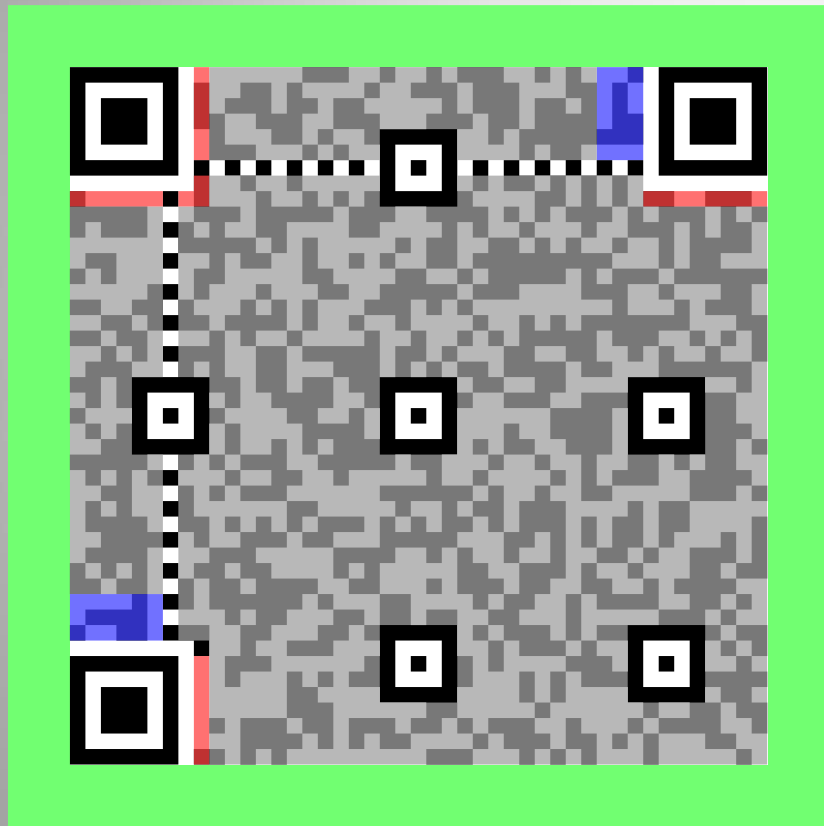






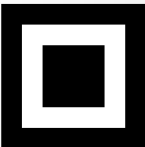
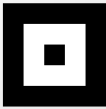


Références



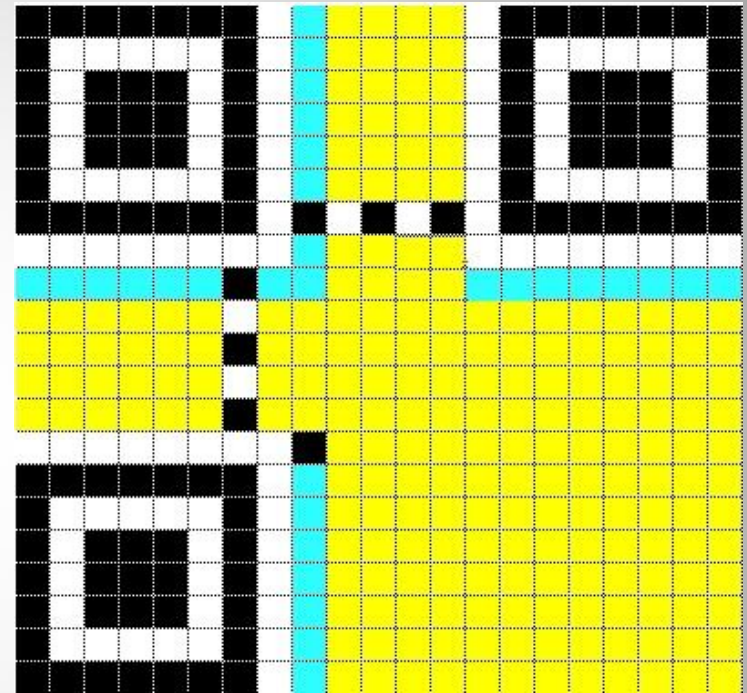
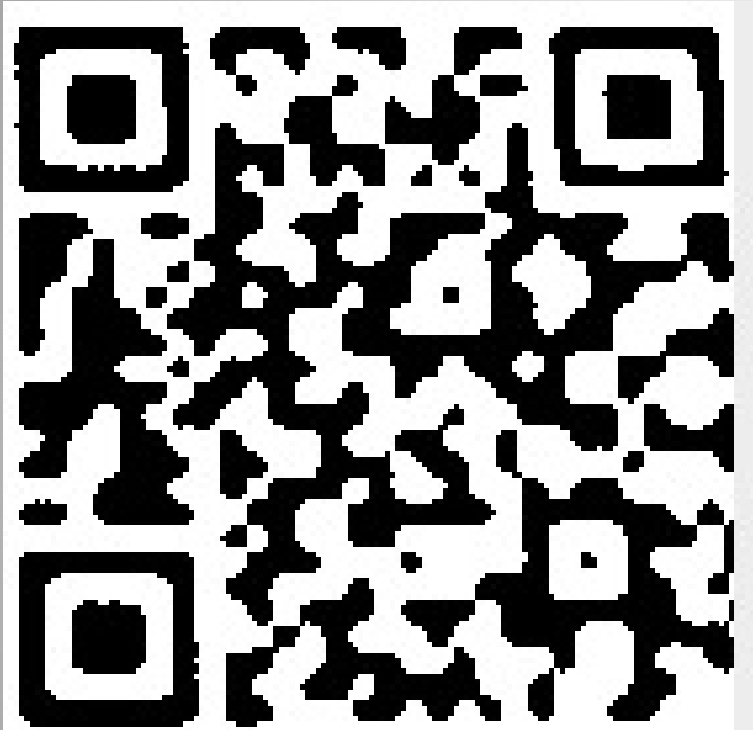
Image QRCode

# Lire le Timing

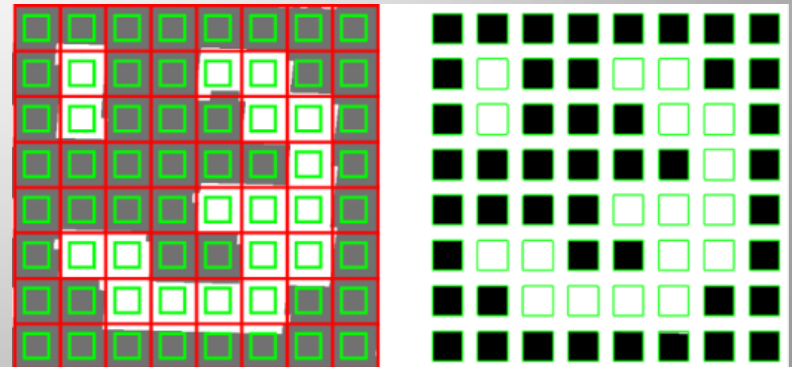


-  1. Version information
-  2. Format information
-  3. Data and error correction keys
-  4. Required patterns
  -  4.1. Position
  -  4.2. Alignment
  -  4.3. Timing
-  5. Quiet zone

# Décodage des bits

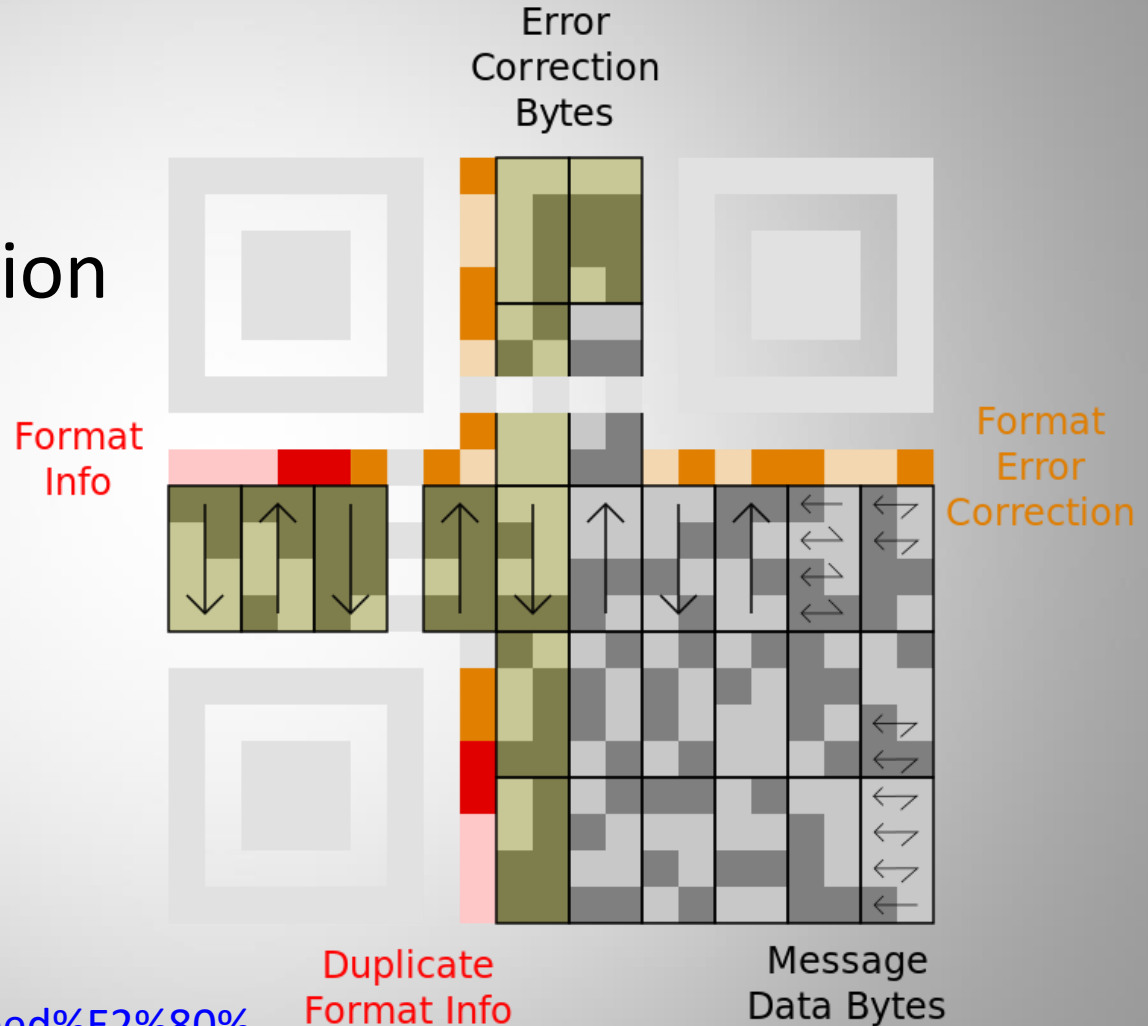


Associer une valeur à  
chaque case: 0 ou 1



# Lire le Format

- Format (1 ->40)
- Code de correction



[https://en.wikiversity.org/wiki/Reed%E2%80%9993Solomon\\_codes\\_for\\_coders](https://en.wikiversity.org/wiki/Reed%E2%80%9993Solomon_codes_for_coders)

# Récupérer les données

Error Correction Level	Level Indicator	Error Correction Bytes	Message Data Bytes
L	01	7	19
M	00	10	16
Q	11	13	13
H	10	17	9

## QR Code Error Correction Capability\*

Level L

Approx 7%

Level M

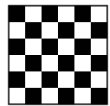
Approx 15%

Level Q

Approx 25%

Level H

Approx 30%



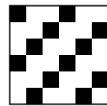
Mask 000  
 $(i + j) \% 2 = 0$



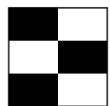
Mask 001  
 $i \% 2 = 0$



Mask 010  
 $j \% 3 = 0$



Mask 011  
 $(i + j) \% 3 = 0$



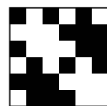
Mask 100  
 $(i/2 + j/3) \% 2 = 0$



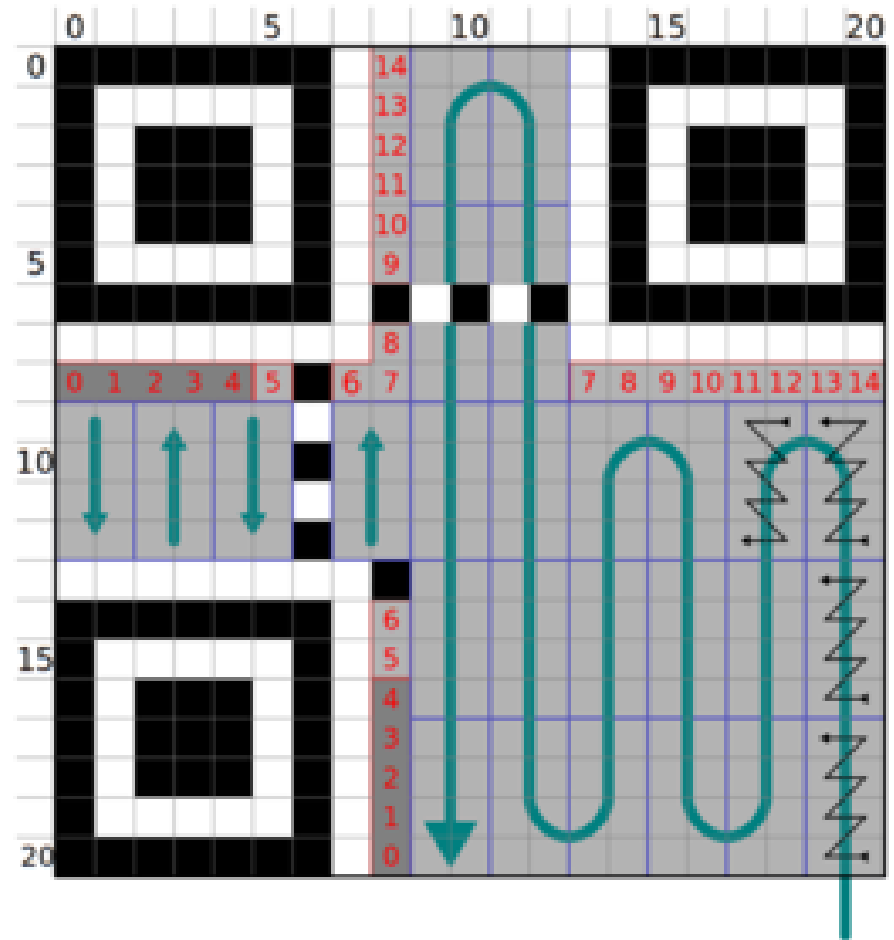
Mask 101  
 $((i*j) \% 2 + (i*j) \% 3) = 0$



Mask 110  
 $((i*j) \% 3 + i*j) \% 2 = 0$



Mask 111  
 $((i*j) \% 3 + i + j) \% 2 = 0$



# Dynamic QRCode

- Le code est statique: Adresse fixe
- Comment le rendre dynamique?
  - Redirection de page (URL Redirection)
  - <http://www.qrstuff.com/>
  - <https://uqr.me/>
  - <http://qrcode.kaywa.com/>
  - <http://fr.qr-code-generator.com/>
  - URL Shortener + link management?
    - goo.gl
    - Bitly.com
    - TinyURL.com

# QR Code liens

- Librairies:
  - Google Barcode Api
    - <https://developers.google.com/vision/android/barcodes-overview>
  - Zxing:
    - <https://github.com/zxing/zxing>
    - <https://zxingnet.codeplex.com/>
  - Zbar:
    - <http://zbar.sourceforge.net/>
    - [best-bar-code-qr-code-scanner-library](#)
  - Payant:
    - <http://www.dynamsoft.com/>



# Unity et Vuforia

- Unity3D/Android... attention aux versions
- Install apk -> charger, exécuter, autoriser
- Facial Landmarks (js):
  - <https://github.com/auduno/clmtrackr>
  - <https://github.com/justadudewhohacks/face-api.js>  
(tensor-flow)
  - Opencv: <https://blog.beautifulinteractions.com/face-detection-and-recognition-with-javascript-9e0b51b1c012>
  - <https://github.com/tehnokv/picojs>
  - <https://www.argonjs.io/>

# Vuforia User Defined Target

- (Finir Vuforia ImageTarget)
- User Defined Target
  - Installation Vuforia Core Samples from asset store
  - Ouvrir UserDefined Target, et autres
- Autre exemple Target :
  - Starwars
- Préparation/continuer votre projet Final

# Vidéo de la démo

- <https://www.youtube.com/watch?v=bRZQn4AcqZg>
- <http://www.instructables.com/id/StarWars-Augmented-Reality-App-for-Beginners/>



# Vuforia

- See **Getting Started with Vuforia in Unity:**  
<https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity-2017-2-beta.html#create-project>

- Ne pas oublier

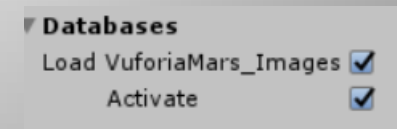
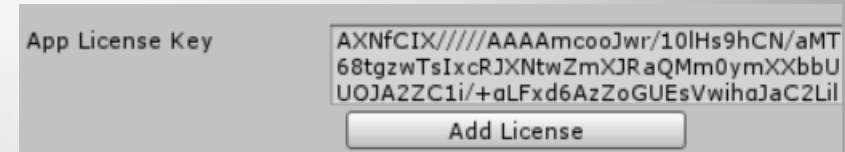
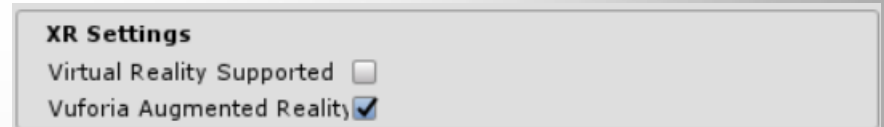
- Dans Build Settings/Player Settings

- L'App License Key

- Activate Vuforia Mars Images

(si demo avec images de vuforia

ou autres database suivant besoin)



# Exercices

- Ajouter des objets fixes dans la scène
- Fond sonore (musique star wars par ex)
- Faire voler un Xwing ou autre
- Mettre une video de maitre Yoda dans cockpit
- Tester sur votre mobile si Android SDK (ios)
- Surprenez-nous...

# Pour la prochaine fois

- **Continuez votre projet**

# Script pour bouger un objet

- <https://www.youtube.com/watch?v=4u177CpPbp8>

`TrackableBehaviour.Status.DETECTED`

To finish